Stat 602 Project Report

Kiegan Rice and Nate Garton

Due ???

Introduction

The goal of this competition was to try to train an effective binary classifier on a small training set with many predictor variables. The criteria used to determine the best solution was the area under the ROC curve generated with predictions on the test set. The training data set had dimensions 250×300, and the test data had dimensions 19750×300.

Exploratory Data Analysis

Data Quality

The quality of the data was pristine. There were no missing values in the test or training data. There was some class imbalance in the training set (roughly 64% of the observations were class 1). A 95%, two-sided confidence interval based on large sample normality assumptions for the population class proportion was $(0.58, 0.70)$. Thus, assuming that we were truly given a random sample from the total data, it seemed unlikely that the classes were truly balanced. However, even at the upper bound of the confidence interval, the class imbalance would not likely be large enough to suggest problems with typical classification models/algorithms.

There were 300 predictor columns given; however, there was no contextual information given about the predictors or the response. There was no information on whether columns represented any particular measured quantity, and each column was simply named $'0', '1', '2', \ldots, '299'$. This makes exploratory data analysis particularly difficult, especially since significant dimension reduction needs to occur to avoid overfitting to noise in the data, and since the training data set only has 250 cases. One clear approach in this scenario is to investigate distributions of each predictor variable, particularly class-conditional distributions since we are working with a classification problem.

Class Conditional Distributions

Because there are so many predictor variables, it is hard to pull useful information from tables of numerical summaries. Figure 1 presents histograms of four summary statistics: minimums, maximums, means, and standard deviations for each column conditional on the class. We can see that the ranges of all of the columns are similar between and within each class. We can also see that columns means for both classes are centered at zero and look fairly Gaussian. There may be one or two "outlier" means in class 0 that are unusually small, but given the number of columns in the data, it was not immediately clear to us that this wasn't simply an artifact of randomly sampled data. Standard deviations for both classes are centered at 1, and most standard deviations for both classes are between 0.9 and 1.1.

The behavior of these summary statistics seems consistent with data where most columns are $N(0, 1)$. With

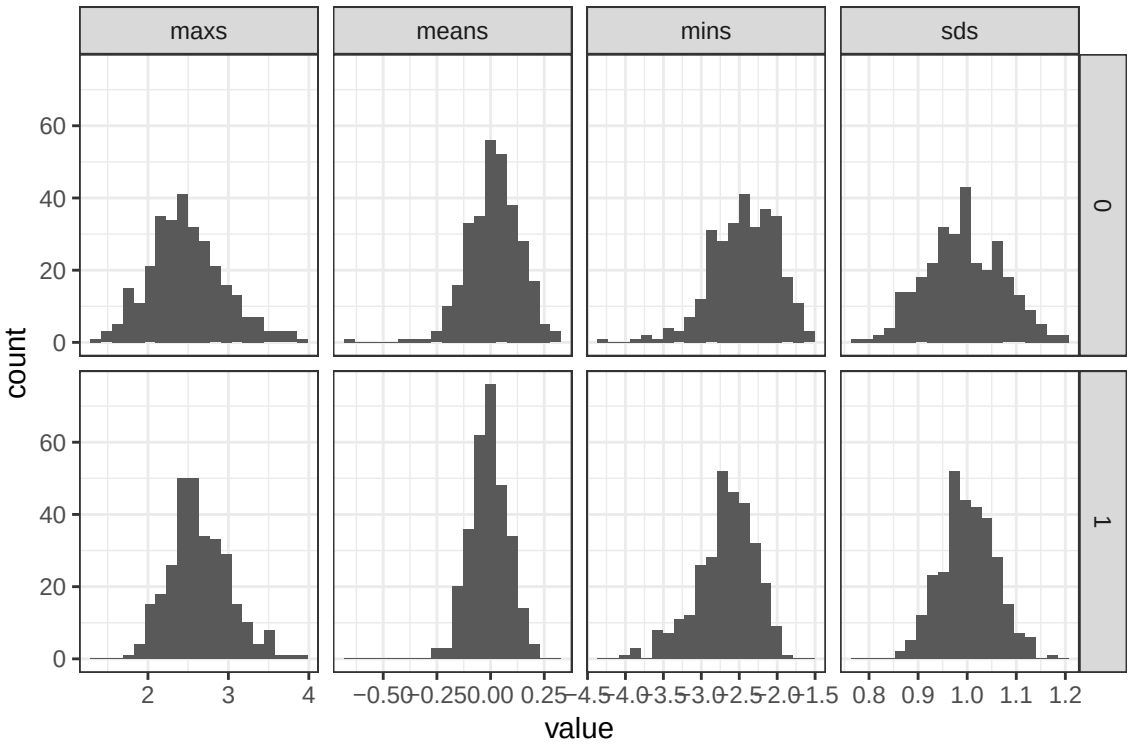

Figure 1: Class conditional histograms for column minimums, maximums, means, and standard deviations. Each row contains the column distribution for these four summary statistics for one of the two classes.

this in mind, we performed the Shapiro-Wilk test for normality on each column for each class. Assuming that the columns are independent for both classes, the distribution of p-values for each class should be approximately uniform. To test this, we performed Kolmogorov-Smirnov (KS) tests for the p-values from the Shapiro-Wilk tests. The two p-values from the (KS) tests were ≈ 0.904 and ≈ 0.997 for classes 0 and 1, respectively. This led us to believe that assuming the normality of the predictor variables was likely reasonable.

If we could also conclude that the predictor variables within each class were independent, then this would allow us to effectively model the class conditional densities. To assess this, we first looked at histograms of the correlations. Figure 2 shows histograms of column correlations within each class. We see that both distributions are centered at essentially zero and appear to have a bell shape. Such a pattern is consistent with the sampling distribution of the correlation between two uncorrelated Gaussian random variables. The distribution of correlations for class 0 appears to be a bit more diffuse, but it is unclear exactly what to make of that. We performed z-tests after using the Fisher transformation for each correlation and then performed the KS test to test whether the p-value distributions for each class were close to uniform. The p-values from these tests were ≈ 0.113 and ≈ 0.905 for class 0 and class 1, respectively. We take these tests with a grain of salt because the correlations used in the test are not independent. However, if there were a handful of high correlations between several predictor variables, we might expect to see more p-values near 0. Finally, the fact that the correlation distribution for class 0 seems more diffuse than class 1 suggests that there might be some subtle differences in correlation structure between the two classes. However, performing a two-sample KS test comparing the correlation distributions between the classes results in a p-value of ≈ 0.251. Again,

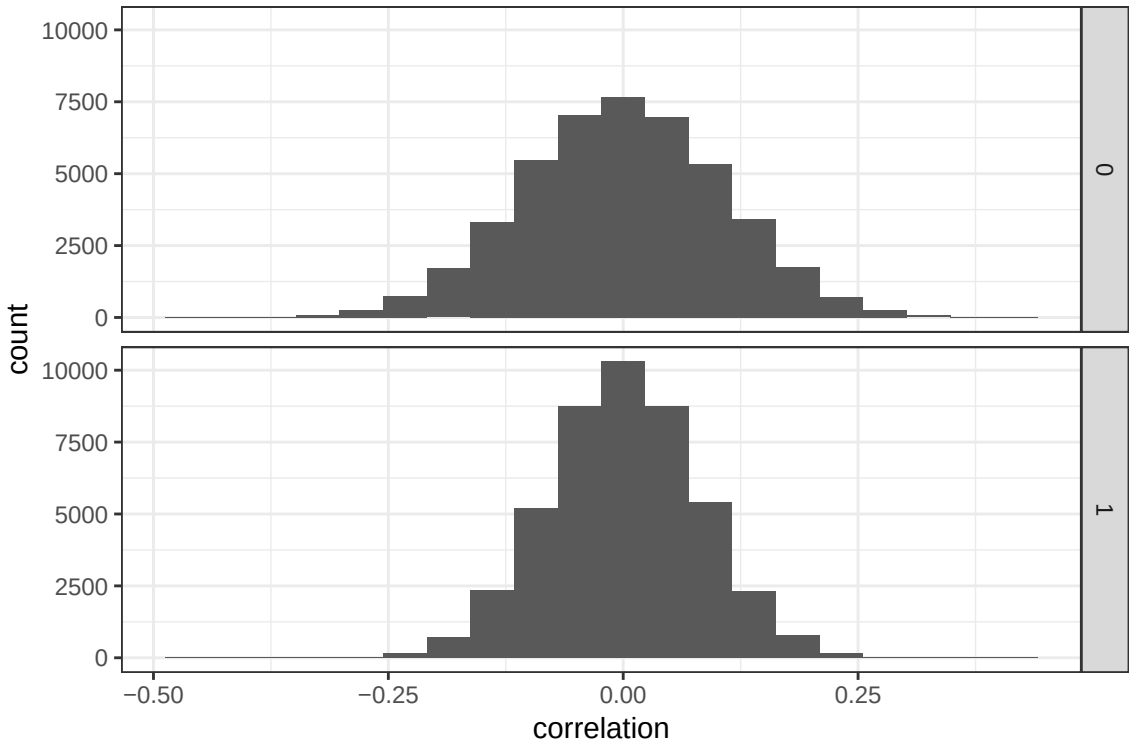

Figure 2: Histograms of column correlations for each class.

the assumption of independence is violated, so we take this with a grain of salt. But, this serves to reinforce the idea that any difference in correlation structure between the two classes is subtle, if it exists. It is also worth commenting that while it is not true that marginally Gaussian, uncorrelated random variables are necessarily independent, if the true joint distribution is not multivariate normal, modeling class conditional densities is a hopelessly challenging task.

Visually investigating class-conditional densities for each column also gives a good look at columns where there may be some slight separation between the two response classes, seen in Figure 3. However, additional statistical tests should be done to assess whether the densities are significantly different, especially since we have such a large number of columns. Some visual separation could be an artifact of random sampling for the training set.

We performed two sample t-tests of the difference in means (assuming equal variance) between each of the corresponding columns from the two classes. If we assume that the columns are all independent (which potentially seems reasonable based), then the Bonferroni correction appropriately controls type-I error without being overly conservative. Two columns have p-values that are significant at the corrected 0.05 type-I error level. Those variables are 33 and 65. Thus, it seems that the separation observed in those two distributions is unlikely to have resulted from sampling variability.

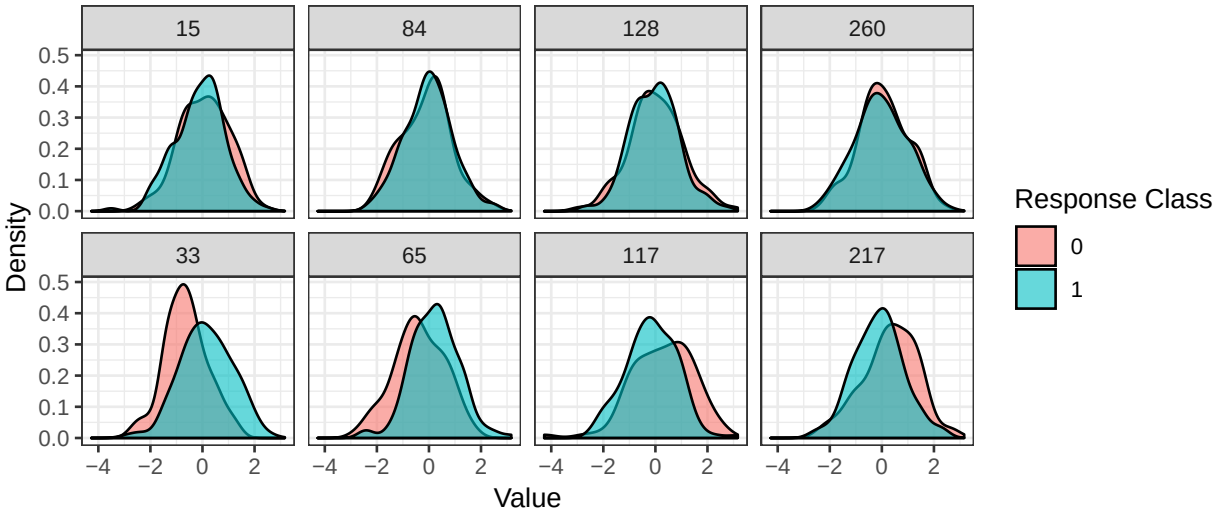

Figure 3: Example of class-conditional densities for several columns in the dataset. The first row shows class-conditional densities that do not demonstrate any separation, which is very common in the dataset. The bottom row demonstrates class-conditional densities observed in the training data which seem to show separation.

Feature Engineering

Creative feature engineering proved particularly difficult for this problem, especially due to the murky nature of the data. While data were pristine in the sense of completeness, the lack of contextual information about predictors makes context-based feature engineering impossible. added a sentence here With that being said, our exploratory data analysis suggests that the class conditional distributions can be modeled well under strong assumptions. In the absence of abundant data, it is critical to take advantage of reasonable assumptions. The large number of predictors also poses the challenge of weeding out noise without removing useful information; dimension reduction plays the first big role in constructing a good set of features.

We have already discussed how two predictor variables appear to be statistically significantly different across classes. However, we would rather do variable selection by optimizing for the criteria upon which we will be judged (AUROC) than using t-test p-values as the selection criteria. Therefore, rather than exclusively using those two variables that has the smallest p-values in our models, we opt for a more traditional type of machine learning variable selection technique. LASSO is a natural choice for dimension reduction here, as it will shrink many predictor parameter estimates to zero in favor of those that more adequately describe the response variable. Furthermore, A logistic regression LASSO model using all 300 predictors was fit using the `glmnet` package, in particular, the `cv.glmnet` function with $\alpha = 1$. Columns with non-zero parameter values in the λ_{1se} model were kept for future the best predictions that we have now come from a LASSO using all of the variables + the LR features, but only column 33 and 65 retained (no LR features) by the lambda.1se model (which is the one used) modeling and feature engineering, while the others were removed from consideration.

Recall that we have argued that it is tentatively reasonable to assume that, conditional on the class, the predictor variable distributions are independent normal random variables with standard deviation 1. In this

case, we should not need to worry about interactions between variables being important for classification, while differences in column distributions corresponding to mean shifts should be identified by the LASSO.

Likelihood Ratios as Features

Our team considered two different approaches to using likelihood ratios as features: a kernel density estimate approach, and an assumed normal distribution approach.

I thought it would be good to have me first describe the kernel density approach, and then you describe the assumed normal distribution approach, and maybe do some plots comparing our two sets of estimated densities. If I can find it in your code, I can just come up with these plots.

Gaussian Likelihood Ratios

Given the assumption that the class conditional distributions of the data are independent Gaussian with standard deviation 1, then the optimal feature is the product of univariate Gaussian likelihood ratios with different means. We estimate the means (conditional on the class) for each of the variables retained by the LASSO using λ_{1se}. Even though it seems reasonable to assume standard deviation 1, we estimate the standard deviations for each Gaussian likelihood as well. Let $f(x|\hat{\mu}_{0i}, \hat{\sigma}_{0i})$ and $f(x|\hat{\mu}_{1i}, \hat{\sigma}_{1i})$ be the Gaussian density functions for variable i for class 0 and 1, respectively, with estimates of the mean (the sample mean) and standard deviation (sample standard deviation) plugged in. We will use x to denote a vector of 300 observed variable values and μ_0, μ_1 and σ_0, σ_1 to denote the vectors of the corresponding means and standard deviations for classes 0 and 1. Denote by I the indicies of the variables (ranging $0, 1, ..., 299$) retained as informative by the LASSO with λ_{1se}. Then, if all variables corresponding to indicies I do, in fact, have different class conditional Gaussian distributions, an estimate of the optimal feature is $\frac{\prod_{i \in I} f(x_i|\hat{\mu}_{1i}, \hat{\sigma}_{1i})}{\prod_{i \in I} f(x_i|\hat{\mu}_{0i}, \hat{\sigma}_{0i})}$. However, supposing that we use a logistic regression as our model to make final predictions, it makes sense to take the log of this feature. This is because

$$\log\left(\frac{P(\text{class} = 1|x)}{P(\text{class} = 0|x)}\right) = \log\left(\frac{f(x|\mu_1, \sigma_1)}{f(x|\mu_0, \sigma_0)}\right) + \log\left(\frac{\pi_1}{\pi_0}\right), \tag{1}$$

where π_0 and π_1 are the overall class probabilities for class 0 and 1. So, we see that the log-odds should be linear in the log-likelihood ratio.

Instead of using this single feature, however, we instead use $|I|$ individual log-likelihood ratios as features. If we use a logistic regression model, we see that using $|I|$ individual log-likelihood ratios as features reduces to the single feature case when all of the regression coefficients are the same. Thus, we allow for slightly more flexible models by using univariate likelihood ratios, but any model class using univariate likelihood ratios contains the single feature model as a submodel.

Clustering

Prediction Methods

With a final set of features, we can try multiple methods for final class predictions.

Random Forest

Conclusions